CLAIMS

1. (Previously Presented) A method for facilitating validation of a system being designed prior to attempting to deploy the system comprising:

receiving, by a system validation computing device configured to facilitate validation of a system being designed prior to attempting to deploy the system, a description of the system being designed to be used in an environment of a data center but not yet deployed to the data center;

receiving, by the system validation computing device, a description of an environment that simulates a target-deployment environment, wherein the target-deployment environment is the data center in which the system is to be deployed; and

using, by the system validation computing device, both of the received descriptions to validate the system against the environment while the system is being designed and prior to attempting to deploy the system to the data center.

2. (Original) A method as recited in claim 1, the description of the system comprising an SDM document.

3. (Original) A method as recited in claim 1, the description of the environment comprising a LIM document.

4. (Canceled)

5. (Canceled)

6. (Previously Presented) One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to perform a method comprising:

accessing an application description that describes an application in the process of being designed to be used in an environment of a data center, by a program running on the one or more processors; and

validating the application, using the application description, against a simulated environment, the environment comprising a description of the data center and prior to deployment to the data center.

7. (Previously Presented) One or more computer readable media as recited in claim 6, the plurality of instructions further causing the processor to:

receive, from a requestor, a request to validate the application; and

return, to the requestor, a result of the validation.

8. (Previously Presented) One or more computer readable media as recited in claim 6, wherein the instructions that cause the one or more processors to validate the application against the simulated environment further cause the one or more processors to perform a method comprising:

selecting a top-level definition from the application description;

generating an appropriate instance, as described by the top-level definition, for an instance space;

selecting an additional definition nested within the top-level definition;

generating an appropriate instance, as described by the additional definition, for the instance space based on whether the selected definition defines an object or a relationship; and

continuing the selection of an additional definition and the generation of an appropriate instance, as described by the additional definition, until instances for all of the definitions nested within the top-level definition have been generated for the instance space.

9. (Previously Presented) One or more computer readable media as recited in claim 6, wherein the instructions that cause the one or more processors to validate the application against the simulated environment further cause the one or more processors to:

identify one or more flows in an instance space, the instance space describing the application;

for each of at least one of the one or more flows:

identify one or more input values for the flow, the input values being obtained from other instances of the instance space; and

generate, based at least in part on the input values, an output value for the flow.

10. (Previously Presented) One or more computer readable media as recited in claim 6, wherein the instructions that cause the one or more processors to validate the

application against the simulated environment further cause the one or more processors to:

identify one or more constraints in an instance space, the instance space describing the application;

check whether the one or more constraints are satisfied; and

return, for each of the one or more constraints, a value indicating whether the constraint is satisfied.

11. (Previously Presented) An apparatus for facilitating validation of a software application being designed to be used in an environment of a data center and prior to attempting to deploy the software application to one area of the data center, the apparatus comprising:

a verifier configured to check one or more documents describing a software application for errors in order for the loader to load;

a loader configured to load the one or more documents describing the software application, the software application being designed when the one or more documents are loaded; and

a simulator configured to simulate an environment of a data center, the environment comprising a description of the data center, and validate the software application against the environment prior to deployment to the data center, and return a result of the validation;

an expansion engine to identify a top-level definition from one of the one or more documents and expand the top-level definition to populate an instance space by instantiating members nested in the top-level definition.

12. (Canceled)

13. (Original) An apparatus as recited in claim 12, further comprising:

a flow engine to identify flows in the instance space, identify the values of inputs to the flows, and setting an output of the flow based on the inputs to the flows.

14. (Original) An apparatus as recited in claim 13, further comprising:

a constraint engine to identify and evaluate constraints in the instance space.

15. (Withdrawn) One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to:

access a document that describes a system being designed to be used in an environment of a data center;

select a top-level definition from the document;

generate an appropriate instance, as described by the top-level definition, for an instance space;

select an additional definition nested within the top-level definition;

generate an appropriate instance, as described by the additional definition, for the instance space based on whether the selected definition defines an object or a relationship; and

continue the selection of an additional definition and the generation of an appropriate instance, as described by the additional definition, until instances for all of the definitions nested within the top-level definition have been generated for the instance space.

16. (Withdrawn) One or more computer readable media as recited in claim 15, wherein the instructions that cause the one or more processors to generate an

appropriate instance, as described by the additional definition, when the selected definition defines a relationship further cause the one or more processors to:

identify a number of relationship instances to create based on a number of source instances and a number of target instances involved in the defined relationship;

create the identified number of relationship instances; and

for each of the created relationship instances, associate source and target instances with the relationship instance.

17. (Withdrawn) One or more computer readable media as recited in claim 16, the selected definition defining a containment relationship that describes that one instance can be contained in another instance.

18. (Withdrawn) One or more computer readable media as recited in claim 16, the selected definition defining a communication relationship that describes an interaction between independently deployed software elements.

19. (Withdrawn) One or more computer readable media as recited in claim 16, the selected definition defining a reference relationship used to capture dependencies between instances.

20. (Withdrawn) One or more computer readable media as recited in claim 16, the selected definition defining a hosting relationship that associates a host with one or more of its guest member instances.

21. (Withdrawn) One or more computer readable media as recited in claim 16, the selected definition defining a delegation relationship that associates communication endpoints of two systems.

22. (Withdrawn) One or more computer readable media as recited in claim 15, wherein the instructions that cause the one or more processors to generate an appropriate instance, as described by the additional definition, when the selected definition defines an object further cause the one or more processors to:

identify a minimum number of occurrences of the object as identified in the selected definition;

identify a number of instances of the selected definition to generate based on the identified minimum number of occurrences and how many instances of the selected definition have already been generated; and

generate the identified number of instances of the selected definition.

23. (Withdrawn) One or more computer readable media as recited in claim 15, wherein the instructions that cause the one or more processors to generate an appropriate instance, as described by the additional definition, when the selected definition defines an object further cause the one or more processors to:

trigger an event that allows a listener to create the appropriate instance as described by the additional information.

24. (Withdrawn) One or more computer readable media as recited in claim 15, wherein the instructions that cause the one or more processors to generate an appropriate instance, as described by the additional definition, when the selected definition defines a relationship further cause the one or more processors to:

trigger an event that allows a listener to create the appropriate instance as described by the additional information.

25. (Withdrawn) One or more computer readable media as recited in claim 15, wherein the instructions are to be executed prior to beginning deployment of the system in the data center.

26. (Withdrawn) One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to:

identify one or more flows in an instance space, the instance space describing a system being designed to be used in an environment of a data center;

for each of at least one of the one or more flows:

identify one or more input values for the flow, the input values being obtained from other instances of the instance space; and

generate, based at least in part on the input values, an output value for the flow.

27. (Withdrawn) One or more computer readable media as recited in claim 26, wherein the instructions that cause the one or more processors to identify one or more input values for the flow further cause the one or more processors to:

identify whether the input values have been assigned yet;

if the input values have been assigned then obtain the input values from the other instances;

if at least one of the input values has not been assigned yet then, for each of the input values that has not been assigned yet:

identify one other flow that sets the input value;

identify one or more input values for the other flow, the input values being obtained from other instances of the instance space; and

generate, based at least in part on the input values, an output value for the other flow.

28. (Withdrawn) One or more computer readable media as recited in claim 26, wherein the instructions that cause the one or more processors to generate, based at least in part on the input values, the output value for the flow further cause the one or more processors to:

identify a set of instructions associated with the flow that can be executed to generate a result;

execute the identified set of instructions; and

use the generated result as the output value for the flow.

29. (Withdrawn) One or more computer readable media as recited in claim 26, the system comprising an application to be deployed in the environment.

30. (Withdrawn) One or more computer readable media as recited in claim 26, the environment comprising a hardware description of a data center.

31. (Withdrawn) One or more computer readable media as recited in claim 26, wherein the instructions are to be executed prior to beginning deployment of the system in the environment.

32. (Withdrawn) One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to:

identify one or more constraints in an instance space, the instance space describing a system being designed to be used in an environment of a data center;

check whether the one or more constraints are satisfied; and

return, for each of the one or more constraints, a value indicating whether the constraint is satisfied.

33. (Withdrawn) One or more computer readable media as recited in claim 32, the one or more constraints including a setting constraint, a relationship constraint, and an object constraint.

34. (Withdrawn) One or more computer readable media as recited in claim 32, wherein the instructions that cause the one or more processors to check whether the one or more constraints are satisfied further cause the one or more processors to, for one of the constraints:

identify a set of instructions associated with the constraint that can be executed to generate a result;

execute the identified set of instructions; and

use the generated result as the value returned indicating whether the constraint is satisfied.

35. (Withdrawn) One or more computer readable media as recited in claim 32, wherein the instructions that cause the one or more processors to check whether the one or more constraints are satisfied further cause the one or more processors to, for one of the constraints:

identify a role and an object definition for a target instance of the constraint;

check whether the role and the object definition of the constraint match the role and the object definition of the target instance; and

generate, based on whether the role and the object definition of the constraint match the role and the object definition of the target instance, the value returned indicating whether the constraint is satisfied.

36. (Withdrawn) One or more computer readable media as recited in claim 35, wherein the instructions that cause the one or more processors to check whether the

one or more constraints are satisfied further cause the one or more processors to, for the one of the constraints:

identify a secondary role and a secondary object definition for the target instance of the constraint;

check whether the secondary role and the secondary object definition of the constraint match the role and the object definition of the target instance; and

generate, based on whether the role and the object definition of the constraint match both the role and the object definition of the target instance and the secondary role and the secondary object definition of the target instance, the value returned indicating whether the constraint is satisfied.

37. (Withdrawn) One or more computer readable media as recited in claim 35, wherein the instructions that cause the one or more processors to check whether the one or more constraints are satisfied further cause the one or more processors to, for the one of the constraints:

evaluate one or more nested constraints for the target instance;

receive one or more return values for the nested constraints, the one or more return values indicating whether the one or more nested constraints are satisfied; and

generate, based on the one or more return values for the nested constraints, the value returned indicating whether the constraint is satisfied.

38. (Withdrawn) One or more computer readable media as recited in claim 32, wherein the instructions that cause the one or more processors to return, for each of the

one or more constraints, a value indicating whether the constraint is satisfied further cause the one or more processors to, for one of the constraints:

if a value indicating that the constraint is not satisfied is to be returned, then check whether an error message is to be generated for the constraint; and

if the error message is to be generated, then generate the error message including information identifying the constraint.

39. (Withdrawn) One or more computer readable media as recited in claim 32, wherein the instructions that cause the one or more processors to check whether the one or more constraints are satisfied further cause the one or more processors to, for one of the constraints:

initialize a match count variable;

identify one or more relationship instances that the target instance of the constraint participates in;

evaluate, for each of the one or more relationship instances, whether the relationship instance satisfies the constraint;

increment the match count variable for each of the one or more relationship instances that satisfies the constraint; and

generate, based on the value of the match count variable after the one or more relationship instances have been evaluated, the value returned indicating whether the constraint is satisfied.

40. (Withdrawn) One or more computer readable media as recited in claim 39, wherein the instructions that cause the one or more processors to evaluate, for each of

the one or more relationship instances, whether the relationship instance satisfies the constraint further cause the one or more processors to, for the one of the constraints:

check whether a relationship definition of the constraint matches a relationship definition of the relationship instance;

check whether a direction of the constraint matches a direction of the relationship instance;

check whether all nested constraints for the relationship instance are satisfied; and

return a value indicating that the constraint is satisfied only if the relationship definition of the constraint matches the relationship definition of the relationship instance, the direction of the constraint matches the direction of the relationship instance, and all nested constraints for the relationship instance are satisfied.

41. (Withdrawn) One or more computer readable media as recited in claim 39, wherein the instructions that cause the one or more processors to evaluate, for each of the one or more relationship instances, whether the relationship instance satisfies the constraint further cause the one or more processors to, for the one of the constraints:

check whether a target object of the constraint matches an instance at the other end of the relationship instance; and

return a value indicating that the constraint is satisfied only if the relationship definition of the constraint matches the relationship definition of the relationship instance, the direction of the constraint matches the direction of the relationship instance, all nested constraints for the relationship instance are satisfied, and the target

object of the constraint matches the instance at the other end of the relationship instance.

42. (Withdrawn) One or more computer readable media as recited in claim 39, wherein the instructions that cause the one or more processors to generate, based on the value of the match count variable after the one or more relationship instances have been evaluated, the value returned indicating whether the constraint is satisfied further cause the one or more processors to, for one of the constraints:

check whether the match count variable is at least a minimum value of the constraint but is not greater than a maximum value of the constraint; and

generate the value returned indicating that the constraint is satisfied if the match count variable is at least the minimum value of the constraint but is not greater than the maximum value of the constraint, otherwise generate the value returned indicating that the constraint is not satisfied.

43. (Withdrawn) One or more computer readable media as recited in claim 32, wherein the instructions are to be executed prior to beginning deployment of the system in the environment.

44. (Previously Presented) An apparatus for facilitating validation of a software application being designed to be used in an environment of a data center and prior to attempting to deploy the software application to one area of the data center, the apparatus comprising:

a verifier configured to check one or more documents describing a software application for errors in order for a loader to load;

a loader configured to load the one or more documents describing the software application;

a simulator configured to simulate an environment of a data center, the environment comprising a description of the data center, to validate the software application against the environment prior to deployment to the data center, and to return a result of the validation;

an expansion engine configured to identify a top-level definition from one of the one or more documents and expand the top-level definition to populate an instance space by instantiating members nested in the top-level definition;

a flow engine configured to identify flows in the instance space, to identify the values of inputs to the flows, and to set an output of the flow based on the inputs to the flows;

a constraint engine configured to identify and evaluate constraints in the instance space;

the apparatus being separate from the data center and the apparatus being comprised, at least in part, of a computer hardware component.

45. (New) A method for facilitating validation of a system being designed prior to attempting to deploy the system, the method comprising:

receiving, from a requestor, a request to validate an application;

receiving, by a system validation computing device configured to facilitate validation of a system being designed prior to attempting to deploy the system, a description of the system being designed to be used in an environment of a data center but not yet deployed to the data center, the description of the system comprising an System Definition Model (SDM) document;

receiving, by the system validation computing device, a description of an environment that simulates a target-deployment environment, the description of the environment comprising a Logical Infrastructure Model (LIM) document, wherein the target-deployment environment is the data center in which the system is to be deployed;

based upon both of the received descriptions, validating, by the system validation computing device, the system against the simulated environment while the system is being designed and prior to attempting to deploy the system to the data center; and

returning, to the requestor, a result of the validating.